

What is claimed is:

1 1: A method comprising:

2 detecting that a first virtual machine is attempting to transmit data to a second
3 virtual machine;

4 mapping a transmitting memory element of the first virtual machine to a shared
5 physical memory element; and

6 mapping a receiving memory element of the second virtual machine to the shared
7 physical memory element.

1 2: The method of claim 1, further including:

2 detecting that the first virtual machine has placed data in the shared physical
3 memory element; and

4 informing the second virtual machine that data is available in the shared physical
5 memory element.

1 3: The method of claim 1, further comprising:

2 detecting if the first virtual machine is attempting to transmit data to a non-virtual
3 machine;

4 dynamically remapping the transmitting memory element of the first virtual
5 machine to a physical device associated with the transmitting memory element.

1 4: The method of claim 3, wherein the transmitting buffer of the first virtual machine is
2 part of a first virtual device; and
3 the receiving buffer of the second virtual machine is part of a second virtual
4 device.

1 5: The method of claim 4, wherein first virtual device and the second virtual device are
2 devices selected from a group including:
3 an Ethernet device,
4 a network interface,
5 an audio device,
6 a storage device, and
7 a video device.

1 6: The method of claim 4, wherein the shared physical memory element is a direct
2 memory access (DMA) buffer.

1 7: The method of claim 1, wherein detecting that a first virtual machine is attempting to
2 transmit data to a second virtual machine includes:
3 monitoring the first virtual machine;
4 comparing the destination of any data transmitted by the first virtual machine to
5 an address associated with the second virtual machine.

1 8: The method of claim 1, wherein detecting that a first virtual machine is attempting to
2 transmit data to a second virtual machine includes:
3 reading a mapping configuration data that specifies default virtual device to
4 physical device mappings;
5 comparing the mapping configuration data for the first virtual machine to the
6 mapping configuration data of the second virtual machine;
7 assuming that the first virtual machine is attempting to transmit data to the second
8 virtual machine, if a transmitting virtual device of the first machine is mapped to the
9 same physical device as the receiving virtual device of the second virtual machine.

1 9: The method of claim 7, wherein mapping a transmitting memory element of the first
2 virtual machine to a shared physical memory element includes:
3 determining if the transmitting memory element is currently mapped to a
4 transmitting memory element of a physical device;
5 if so, unmapping of the transmitting memory element from the transmitting
6 memory element of the physical device; and
7 remapping the transmitting memory element of the first virtual machine to the
8 shared physical memory element.

1 10: The method of claim 9, further including:

2 detecting that the second virtual machine is attempting to transmit data to the first
3 virtual machine;

4 mapping a transmitting memory element of the second virtual machine to the
5 shared physical memory element; and

6 mapping a receiving memory element of the first virtual machine to the shared
7 physical memory element.

1 11: An article comprising:

2 a machine accessible medium having a plurality of machine accessible instructions,
3 wherein when the instructions are executed, the instructions provide for:

4 detecting that a first virtual machine is attempting to transmit data to a second
5 virtual machine;

6 mapping a transmitting memory element of the first virtual machine to a shared
7 physical memory element; and

8 mapping a receiving memory element of the second virtual machine to the shared
9 physical memory element.

1 12: The article of claim 11, further including instructions providing for:

2 detecting that the first virtual machine has placed data in the shared physical
3 memory element; and

4 informing the second virtual machine that data is available in the shared physical
5 memory element.

1 13: The article of claim 11, further comprising instructions providing for:
2 detecting if the first virtual machine is attempting to transmit data to a non-virtual
3 machine:
4 dynamically remapping the transmitting memory element of the first virtual
5 machine to a physical device associated with the transmitting memory element.

1 14: The article of claim 13, wherein the transmitting buffer of the first virtual machine is
2 part of a first virtual device; and
3 the receiving buffer of the second virtual machine is part of a second virtual
4 device.

1 15: The article of claim 14, wherein first virtual device and the second virtual device are
2 devices selected from a group including:
3 an Ethernet device,
4 a network interface,
5 an audio device,
6 a storage device, and
7 a video device.

1 16: The article of claim 14, wherein the shared physical memory element is a direct
2 memory access (DMA) buffer.

1 17: The article of claim 11, wherein the instructions provide for detecting that a first
2 virtual machine is attempting to transmit data to a second virtual machine includes
3 instructions providing for:
4 monitoring the first virtual machine;
5 comparing the destination of any data transmitted by the first virtual machine to
6 an address associated with the second virtual machine.

1 18: The article of claim 11, wherein the instructions provide for detecting that a first
2 virtual machine is attempting to transmit data to a second virtual machine includes
3 instructions providing for:
4 reading a mapping configuration data that specifies default virtual device to
5 physical device mappings;
6 comparing the mapping configuration data for the first virtual machine to the
7 mapping configuration data of the second virtual machine;
8 assuming that the first virtual machine is attempting to transmit data to the second
9 virtual machine, if a transmitting virtual device of the first machine is mapped to the
10 same physical device as the receiving virtual device of the second virtual machine.

1 19: The article of claim 17, wherein the instructions provide for mapping a transmitting
2 memory element of the first virtual machine to a shared physical memory element
3 includes instructions providing for:
4 determining if the transmitting memory element is currently mapped to a
5 transmitting memory element of a physical device;
6 if so, unmapping of the transmitting memory element from the transmitting
7 memory element of the physical device; and
8 remapping the transmitting memory element of the first virtual machine to the
9 shared physical memory element.

1 20: The article of claim 19, further including instructions providing for:
2 detecting that the second virtual machine is attempting to transmit data to the first
3 virtual machine;
4 mapping a transmitting memory element of the second virtual machine to the
5 shared physical memory element; and
6 mapping a receiving memory element of the first virtual machine to the shared
7 physical memory element.

1 21: A virtual machine manager comprising:
2 a cross-talk detector to detect if a first virtual machine is attempting to transmit

3 data to a second virtual machine; and
4 a dynamic memory remapper to, if instructed by the cross-talk detector, map a
5 first virtual memory of the first virtual machine to a second virtual memory of the second
6 virtual machine via a shared physical memory element.

1 22: The virtual machine manager of claim 21, wherein the dynamic memory remapper is
2 capable of:

3 mapping a transmitting memory element of the first virtual machine to a shared
4 physical memory element; and

5 mapping a receiving memory element of the second virtual machine to the shared
6 physical memory element.

1 23: The virtual machine manager of claim 22, wherein the cross-talk detector is further
2 capable of:

3 detecting that the first virtual machine has placed data in the shared physical
4 memory element; and

5 informing the second virtual machine that data is available in the shared physical
6 memory element.

1 24: The virtual machine manager of claim 22, wherein the cross-talk detector is further
2 capable of detecting if the first virtual machine is attempting to transmit data to a non-

3 virtual machine; and
4 the dynamic memory remapper is further capable of dynamically remapping the
5 transmitting memory element of the first virtual machine to a physical device associated
6 with the transmitting memory element.

1 25: The virtual machine manager of claim 24, wherein the cross-talk detector is further
2 capable of monitoring the first and second memories wherein the memories are part of
3 virtual devices selected from a group including:

4 an Ethernet device,
5 a network interface,
6 an audio device,
7 a storage device, and
8 a video device.

1 26: The virtual machine manager of claim 25, wherein the shared physical memory
2 element is a direct memory access (DMA) buffer.

1 27: The virtual machine manager of claim 26, wherein the cross-talk detector is capable
2 of:
3 monitoring the first virtual machine;

4 comparing the destination of any data transmitted by the first virtual machine to
5 an address associated with the second virtual machine.

1 28: The virtual machine manager of claim 22, wherein the dynamic memory remapper is
2 capable of:

3 determining if the transmitting memory element is currently mapped to a
4 transmitting memory element of a physical device;

5 if so, unmapping of the transmitting memory element from the transmitting
6 memory element of the physical device; and

7 remapping the transmitting memory element of the first virtual machine to the
8 shared physical memory element.

1 29: The virtual machine manager of claim 28, wherein the cross-talk detector is capable
2 of detecting that the second virtual machine is attempting to transmit data to the first
3 virtual machine; and

4 wherein the dynamic memory remapper is capable of mapping a transmitting memory
5 element of the second virtual machine to the shared physical memory element; and

6 mapping a receiving memory element of the first virtual machine to the shared
7 physical memory element..

1 30: The virtual machine manager of claim 21, wherein the first virtual memory element
2 of the first virtual machine to a second virtual memory element of the second virtual
3 machine are not identical but share substantially similar characteristics.

1 31: A system comprising:
2 a first virtual machine, having a first virtual device that includes a first virtual memory
3 element;
4 a second virtual machine, having a second virtual device that includes a second virtual
5 memory element;
6 a shared physical memory element; and
7 a virtual machine manager, having
8 a cross-talk detector to detect if a first virtual machine is attempting to transmit
9 data to a second virtual machine; and
10 a dynamic memory remapper to, if instructed by the cross-talk detector, map a
11 first virtual memory element of the first virtual machine to a second virtual memory
12 element of the second virtual machine via a shared physical memory element.

1 32: The system of claim 31, wherein the dynamic memory remapper is capable of:
2 mapping a transmitting memory element of the first virtual machine to a shared
3 physical memory element; and
4 mapping a receiving memory element of the second virtual machine to the shared
5 physical memory element.

1 33: The system of claim 32, wherein the cross-talk detector is further capable of:
2 detecting that the first virtual machine has placed data in the shared physical
3 memory element; and
4 informing the second virtual machine that data is available in the shared physical
5 memory element.

1 34: The system of claim 32, wherein the cross-talk detector is further capable of
2 detecting if the first virtual machine is attempting to transmit data to a non-virtual
3 machine; and
4 the dynamic memory remapper is further capable of dynamically remapping the
5 transmitting memory element of the first virtual machine to a physical device associated
6 with the transmitting memory element.

1 35: The system of claim 34, wherein the cross-talk detector is further capable of
2 monitoring the first and second memories wherein the memories are part of virtual
3 devices selected from a group including:
4 an Ethernet device,
5 a network interface,
6 an audio device,

7 a storage device, and
8 a video device.

1 36: The system of claim 35, wherein the shared physical memory element is a direct
2 memory access (DMA) buffer.

1 37: The system of claim 36, wherein the cross-talk detector is capable of:
2 monitoring the first virtual machine;
3 comparing the destination of any data transmitted by the first virtual machine to
4 an address associated with the second virtual machine.

1 38: The system of claim 32, wherein the dynamic memory remapper is capable of:
2 determining if the transmitting memory element is currently mapped to a
3 transmitting memory element of a physical device;
4 if so, unmapping of the transmitting memory element from the transmitting
5 memory element of the physical device; and
6 remapping the transmitting memory element of the first virtual machine to the
7 shared physical memory element.

1 39: The system of claim 38, wherein the cross-talk detector is capable of detecting that
2 the second virtual machine is attempting to transmit data to the first virtual machine; and
3 wherein the dynamic memory remapper is capable of mapping a transmitting memory
4 element of the second virtual machine to the shared physical memory element; and
5 mapping a receiving memory element of the first virtual machine to the shared
6 physical memory element.

1 40: The system of claim 31, wherein the first virtual memory element of the first virtual
2 machine to a second virtual memory element of the second virtual machine are not
3 identical but share substantially similar characteristics.

1 41: A method of communicating between two virtual machines utilizing a virtual
2 machine manger comprising:
3 detecting that a first virtual machine, having a first virtual network interface, is
4 attempting to transmit data to a second virtual machine, second virtual network interface,
5 via the virtual network interfaces;
6 mapping a transmitting memory element of the first virtual network interface to a
7 direct memory access buffer; and
8 mapping a receiving memory element of the second virtual network interface to
9 the direct memory access buffer.

1 42: The method of claim 41, further including:

2 detecting that the first virtual machine has placed data in the direct memory

3 access buffer; and

4 informing the second virtual machine that data is available in the direct memory

5 access buffer.

1 43: The method of claim 41, further comprising the first virtual machine:

2 placing at least one packet into the direct memory access buffer; and

3 moving the tail register of the first virtual network interface to indicate how many

4 packets where written to the direct memory access buffer.

1 44: The method of claim 43, further comprising the virtual machine manager:

2 moving the receive descriptor head register of the second network interface by the

3 number of packets written to the direct memory access buffer;

4 updating the status of the second network interface to indicate that a packet has

5 been received;

6 sending a receive interrupt to the second virtual machine.

1 45: The method of claim 44, further comprising the second virtual machine:

2 reading the data from the direct memory access buffer.

1 46: The method of claim 45, further comprising, the virtual machine manager:
2 detecting that the second virtual machine has read the data from the direct
3 memory buffer;
4 updating the status of the first network interface to indicate that the data has been
5 received; and
6 injecting a transmit complete interrupt to the first virtual machine.

1 47: The method of claim 41, wherein detecting that a first virtual machine is attempting
2 to transmit data to a second virtual machine includes:
3 monitoring the first virtual machine;
4 comparing the destination of any data transmitted by the first virtual machine to
5 an address associated with the second virtual machine.

1 48: A method comprising:
2 detecting that a first virtual machine is configured to transmit data to a second
3 virtual machine;
4 statically mapping a transmitting memory element of the first virtual machine to a
5 shared physical memory element; and
6 statically mapping a receiving memory element of the second virtual machine to
7 the shared physical memory element.

1 49: The method of claim 48, further comprising:

2 statically mapping a receiving memory element of the first virtual machine to a
3 second shared physical memory element; and

4 statically mapping a transmitting memory element of the second virtual machine
5 to the second shared physical memory element.

1 50: The method of claim 48, wherein detecting that a first virtual machine is configured
2 to transmit data to a second virtual machine is done when the first virtual machine is
3 started.

1 51: The method of claim 50, wherein detecting that a first virtual machine is configured
2 to transmit data to a second virtual machine includes reading a configuration file that
3 explicitly denotes that the first and second virtual machines are virtually coupled.

1 52: The method of claim 50, wherein detecting that a first virtual machine is configured
2 to transmit data to a second virtual machine includes reading a configuration file that
3 implicitly denotes that the first and second virtual machines are virtually coupled.

1 53: The method of claim 50, wherein the shared physical memory element comprises a
2 direct access memory buffer.

1 54: The method of claim 53, wherein the virtual memory elements of the first and second
2 virtual machines are part of virtual devices selected from a group of virtual devices
3 comprising:
4 an Ethernet device;
5 a network device;
6 an audio device; a storage device; and
7 a video device.

1 55: An article comprising:
2 a machine accessible medium having a plurality of machine accessible instructions,
3 wherein when the instructions are executed, the instructions provide for:
4 detecting that a first virtual machine is configured to transmit data to a second
5 virtual machine;
6 statically mapping a transmitting memory element of the first virtual machine to a
7 shared physical memory element; and
8 statically mapping a receiving memory element of the second virtual machine to
9 the shared physical memory element.

1 56: The article of claim 55, further comprising instructions providing for:
2 statically mapping a receiving memory element of the first virtual machine to a
3 second shared physical memory element; and
4 statically mapping a transmitting memory element of the second virtual machine
5 to the second shared physical memory element.

1 57: The article of claim 55, wherein the instructions providing for detecting that a first
2 virtual machine is configured to transmit data to a second virtual machine are executed
3 when the first virtual machine is started.

1 58: The article of claim 57, wherein the instructions providing for detecting that a first
2 virtual machine is configured to transmit data to a second virtual machine includes
3 instructions providing for reading a configuration file that explicitly denotes that the first
4 and second virtual machines are virtually coupled.

1 59: The article of claim 57, wherein the instructions providing for detecting that a first
2 virtual machine is configured to transmit data to a second virtual machine includes
3 instructions providing for reading a configuration file that implicitly denotes that the first
4 and second virtual machines are virtually coupled.

1 60: The article of claim 57, wherein the instructions provide for the shared physical
2 memory element comprising a direct access memory buffer.

1 61: The article of claim 60, wherein the virtual memory elements of the first and second
2 virtual machines are part of virtual devices selected from a group of virtual devices
3 comprising:
4 an Ethernet device;
5 a network device;
6 an audio device; a storage device; and
7 a video device.